

## **REMARKS**

This is a full and timely response to the outstanding non-final Office Action mailed August 22, 2006. Reconsideration and allowance of the application and pending claims are respectfully requested.

### **I. Drawing Amendments**

Replacement sheets have been provided for Figures 1-4B to correct the font of the figure labels. No new matter has been added.

### **II. Claim Rejections - 35 U.S.C. § 101**

Claims 1-29 have been rejected under 35 U.S.C. § 101 because, in the Examiner's opinion, none of the claims disclose a "useful, tangible, and concrete result." Applicant disagrees.

As stated in the Background section of the patent application:

In order to facilitate debugging of the program at issue, it is necessary to determine whether the discovered bug is a new bug or a bug that was identified earlier, and is therefore already logged in the bug database. Although the debugger or another human being can manually scan through the bug database to determine if a bug record already exists for the discovered bug, *this process is inefficient and is highly susceptible to human error. For instance, it may be difficult to identify an already-recorded bug if the number of bugs contained in the bug database and/or the number of entries in the bug records are large.*

*Applicant's specification*, page 1, line 5 to page 2, line 2 (emphasis added). Accordingly, it is difficult for human debuggers to determine whether a discovered bug is one that has already been

identified. Applicant's claimed inventions greatly simplify that determination process. As further provided in Applicant's specification:

Disclosed are systems and methods for identifying similar bugs to aid a user (e.g., debugger) in determining whether a bug in question is a new bug or a bug that has already been identified and, therefore, is logged in a bug database. In some embodiments, a system and method can be used to automatically generate a derivative database based upon information contained in a bug database, and *automatically make a probability determination as to the likelihood that one or more bugs of the bug database are the same as a bug in question. In such a case, the user can be provided with information that the user can use to make his or her own determination as to whether the bug in question is a new or a previously-identified bug.*

*Applicant's specification*, page 3, lines 6-14 (emphasis added). Therefore, using Applicant's invention, a human debugger can be provided with information that is indicative of the probability of a discovered bug being a previously identified bug, greatly simplifying the bug identification process for the debugger.

In view of the above, Applicant submits that the claimed inventions do provide a result that is "useful, tangible, and concrete." Specifically, the tangible, concrete information (i.e., the "probability determination") generated by the inventions are useful to the human debugger in deciding whether a bug is one that has or has not already been encountered. Therefore, Applicant submits that the claims are proper under 35 U.S.C. § 101 and requests that the rejection be withdrawn.

Claims 15-24 have been rejected under 35 U.S.C. § 101 as being drawn to non-statutory subject matter. In particular, the Examiner argues that those claims “lack tangible hardware, memory, input/outs, and sources.” Applicant disagrees.

Claims 15-24 each recite various “means” for doing something. Applicant notes that means-plus-function recitations must be interpreted with reference to the specification. Applicant further notes that Applicant has described various hardware in the specification. Even assuming that one or more of those means comprise software, the recited “means” can properly be interpreted as further comprising a processing device, such as processing device 102 in Figure 1, that executes the software. Accordingly, Applicant submits that claims 15-24 comprise the necessary “lack tangible hardware, memory, input/outs, and sources” under 35 U.S.C. § 101 and requests that the rejection be withdrawn.

### **III. Abstract Objection**

The abstract of the disclosure has been objected to because of the inclusion of legal terminology. Through this Response, all such legal language has been removed. In view of that amendment, Applicant respectfully requests that the objection be withdrawn.

### **IV. Claim Rejections - 35 U.S.C. § 102(e)**

Claims 1-29 have been rejected under 35 U.S.C. § 102(e) as being anticipated by *Hines* (U.S. Pat. No. 6,859,893). Applicant respectfully traverses this rejection.

It is axiomatic that “[a]nticipation requires the disclosure in a single prior art reference of each element of the claim under consideration.” *W. L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 1554, 220 U.S.P.Q. 303, 313 (Fed. Cir. 1983). Therefore, every claimed feature of the

claimed invention must be represented in the applied reference to constitute a proper rejection under 35 U.S.C. § 102(e).

In the present case, not every feature of the claimed invention is represented in the Hines reference. Applicant discusses the Hines reference and Applicant's claims in the following.

#### **A. The Hines Disclosure**

Hines discloses a system and method for automatic computer system analysis. As described by Hines, when a problem occurs on a client computer system, such as a system crash, the computer system creates a core file comprising a memory image of the state of the computer prior to the crash. *Hines*, column 7, lines 25-34. The core file is then provided to a service guru tool 150 that automatically analyzes the core file to produce a “report” indicating identified problems and corrective actions. *Hines*, column 8, lines 27-33.

The service guru tool 150 analyzes the core file by searching for applicable patches for correcting identified bugs in a repository 171 comprising a collection of “phases” and scripts (i.e., short programs) that can be run to conduct the analysis. *Hines*, column 8, lines 44-56. One script is provided for each unique problem. *Hines*, column 8, lines 60-64. The service guru tool 150 builds a list of phases from the repository 171 and runs the scripts of those phases. *Hines*, column 9, lines 58-62; column 10, lines 22-23.

Hines describes running of the scripts in relation to Figure 4. In an initial phase, the core file is parsed to create an intermediate context file. *Hines*, column 12, lines 36-48. Next, a check for bad patches is performed. *Hines*, column 12, lines 57-60. Next, field information notice compliance and FCO compliance are performed. *Hines*, column 12, lines 12, lines 60-64. Next, a hardware scan is performed by scanning hardware error files for matches with the client computer

system. *Hines*, column 12, lines 64-67. Next, a software error scan is performed on the input information followed by an infodoc check, an faq check, and an srdb check. *Hines*, column 13, lines 1-6. Next, performed are a stb/white paper check, retrieval of all down revision patches, and a security issue check. *Hines*, column 13, lines 6-9. Next, a heath check is performed. *Hines*, column 13, lines 15-18. Next, storage related checks are performed. *Hines*, column 13, lines 18-24. Next, a “coretool” tool is run on the client computer system. *Hines*, column 13, lines 25-28. Next, a series of platform specific tests are run. *Hines*, column 13, lines 28-30. Next, a series of performance related checks are performed. *Hines*, column 13, lines 30-33. Next, a bug analysis phase is performed. In that phase, the service guru tool 150 performs “screening of the identified bug files 172 in the knowledge database server 170 to rule out or eliminate bugs that cannot apply to the target computer system 110 based upon system parameters such as loaded packages 116, versions of operating systems or applications, patch levels, and other factors noted in the bug files 172.” *Hines*, column 13, lines 34-41.

After the each phase of the analysis has been completed, a report is generated. *Hines*, column 14, lines 58-60. As described by Hines:

In a preferred embodiment, the report combines and arranges the intermediate reports from each phase and displays the report using text format with embedded URLs on the user interface 144. For example, the results may be ordered by the order the checks and phases were completed or by the severity of the problems or bugs identified. The output report preferably is viewable with standard interface applications such as Netscape.TM., dtmail, and the like. Each report preferably includes a recommended action or service guru comment, a type (i.e., proactive, reactive, or informational), and a severity. Additionally, the output report

preferably is configured such that the report information is searchable because of the large number of problems/issues that may match for a given system 110.

*Hines*, column 15, lines 9-22.

In view of the above, Hines describes a system and method for analyzing encountered problems. Significantly however, Hines says nothing of generating “tokens” associated with known bugs or bugs in question, or determining a statistical probability as to whether a bug in question is one of the known bugs, or the various steps involved in performing such statistical probability analysis.

## **B. Discussion of the Rejections**

As an initial matter, Applicant objects to the rejections as being too vague to clearly reveal the basis for the Examiner’s rejection. In particular, the Examiner provides no explanation as to what Hines teaches or how those teachings are the same or equivalent to each of Applicant’s limitations in claims 1-29. Instead, the Examiner merely block copies each of Applicant’s claim limitations, states that Hines teaches that limitation, and cites a portion of the Hines disclosure without discussion. Although such a method would be adequate if the various cited portions of the Hines reference explicitly taught the various limitations, Applicant notes that, more often than not, the cited portions do not comprise a description of what Applicant is claiming. For example, Applicant describes “tokens” in various claims. None of the portions of the Hines disclosure relied upon in rejections, however, describe a “token”. As another example, Applicant recites in some claims that the “Bayes Theorem” is used in the probability determination, although the Hines disclosure does not even contain the term “Bayes” and further does not describe any form of statistical analysis.

In view of the above, Applicant requests that the Examiner provide an explanation as to how each of the cited portions of the Hines references actually teaches Applicant's explicitly recited limitations. If the Examiner refuses to do so, the Examiner will deny Applicant a full and fair hearing as to the patentability of Applicant's claims. As provided in MPEP 706.07, "[t]he Examiner should never lose sight of the fact that in every case the applicant is entitled to a full and fair hearing, and that a clear issue between applicant and examiner should be developed, if possible, before appeal."

Given that the Office Action provides no explanation as to how Hines actually anticipates each and every limitation of Applicant's claims, Applicant will not discuss each claim separately as such a discussion would result in a multiplicity of phrases such as "contrary to that alleged in the Office Action, column \_\_, lines \_\_-\_\_ do not teach . . ." Applicant provides a brief discussion of claim 1 in the following, however, as representative of the deficiencies of the Hines reference as a 102 reference as to Applicant's claims.

Applicant's claim 1 provides as follows (emphasis added):

1. A method for identifying similar bugs, comprising:  
generating a database that contains database tokens that relate to identified bugs;  
generating input tokens associated with a bug in question;  
scanning the database for occurrences of the input tokens; and  
determining an overall statistical probability as to whether the identified bugs are the same as the bug in question.

Regarding claim 1, Hines does not teach generating a database that contains database “tokens” that relate to identified bugs. Contrary to that alleged in the Office Action, column 8, lines 50-56 describe no “tokens” or any elements equivalent to Applicant’s claimed tokens.

Hines further does not teach “generating input tokens associated with a bug in question”. Contrary to that alleged in the Office Action, column 13, line 62 to column 14, lines 2 describes no “tokens” or any elements equivalent to Applicant’s claimed tokens.

Hines further does not teach “scanning the database for occurrences of the input tokens” given that Hines discloses no “tokens” at all.

Hines further does not teach “determining an overall statistical probability as to whether the identified bugs are the same as the bug in question”. Column 14, lines 17-31 of the Hines reference say nothing of any statistical probability determination.

## **V. New Claims**

Claims 30-36 have been added into the application through this Response. Applicant respectfully submits that these new claims describe an invention novel and unobvious in view of the prior art of record and, therefore, respectfully requests that these claims be held to be allowable.

## CONCLUSION

Applicant respectfully submits that Applicant's pending claims are in condition for allowance. Favorable reconsideration and allowance of the present application and all pending claims are hereby courteously requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned attorney at (770) 933-9500.

Respectfully submitted,



\_\_\_\_\_  
David R. Risley  
Registration No. 39,345

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, postage prepaid, in an envelope addressed to: Assistant Commissioner for Patents, Alexandria, Virginia 22313-1450, on

10-4-06

May Meehan  
Signature